"Express Mail" Mailing Label No. **EL739930235US**

**PATENT APPLICATION**
**ATTORNEY DOCKET NO. OR01-04201**

# EFFICIENT COLLATION ELEMENT STRUCTURE FOR HANDLING LARGE NUMBERS OF CHARACTERS

**Inventors:** Ching-Lan Ho and Jianping Yang

## BACKGROUND

### Field of the Invention

The present invention relates to the process of indexing and sorting data within a database system. More specifically, the present invention relates to a method and an apparatus for providing an efficient collation element structure for encoding sorting weights for a large number of characters.

### Related Art

One challenge in developing a database system is to support sorting for different languages. For example, some databases allow Japanese customers to specify the sorting method as "Japanese", while French customers specify the

sorting method as "French". However, with the worldwide deployment of Internet technologies, it is becoming increasingly important for companies to provide multi-lingual capabilities in their software in order to expand their businesses globally.

As the data that is stored in databases becomes increasingly more multi-lingual, database users are increasingly more interested in using a single sort whose order is good for most languages.

Unfortunately, there can be thousands of different characters in a multi-lingual sort, and this causes the data structures involved in performing the multi-lingual sort to consume a lot of memory. Moreover, it can be hard to compact these data structures without degrading performance.

Multi-lingual sorting is typically accomplished by converting strings of characters into corresponding strings of collation elements (these strings are also known as sorting keys), and then comparing the strings of collation elements to perform the sorting operation. This conversion process is typically accomplished by looking up characters in a collation weight table that contains a corresponding collation weight for each character.

Unicode Technical Report No. 10 (from the Unicode Consortium) specifies a collation element structure that includes a 16-bit primary weight value followed by an eight-bit secondary weight value and an eight-bit tertiary weight value. The primary weight value identifies a character, while the secondary weight value specifies an accent on the character, and the tertiary weight value specifies case information (and possibly related punctuation) for the character. For example, the primary weight value may specify that a character is an "a", while the secondary value specifies that the character has an accent "ä", and the tertiary value specifies that the character is upper case "Ä".

Note that a comparison function typically compares the primary weights first. If the primary weights match, the comparison function compares the secondary weights. If both primary and secondary weights match, the comparison function compares the tertiary weights.

Note that the 16-bit primary weight value specified by Unicode Technical Report No. 10 can only encode 65,536 different characters. However, it is becoming necessary to provide more that 65,536 characters. This can be accomplished by increasing the size of the primary weight value to 32 bits (4 bytes). However, increasing the size of the primary weight value from 16 to 32 bits has a number of disadvantages: (1) more memory is required to build a linguistic index to support the 32-bit primary weight value; (2) access time to the linguistic index is increased; (3) more memory is required to store strings of collation elements; and (4) more computational operations are required to compare sorting keys.

What is needed is a method and an apparatus for using an efficient collation element structure that can handle a large number of characters without the above-mentioned problems.

## SUMMARY

One embodiment of the present invention provides a system for facilitating use of a collation element that supports a large number of characters. The system operates by receiving the collation element and reading a primary weight value from a primary weight field within the collation element. If the primary weight value falls within a reserved set of values, the system reads an additional portion of the primary weight value from both a secondary weight field and a tertiary weight field within the collation element. On the other hand, if the primary weight value is not within the reserved set of values, the system reads a secondary

weight value from the secondary weight field, and also reads a tertiary weight value from the tertiary weight field.

In one embodiment of the present invention, if the primary weight value falls within a reserved set of values, the system sets the secondary weight value to a secondary default value, and sets the tertiary weight value to a tertiary default value.

In one embodiment of the present invention, the collation element adheres to the Unicode standard.

In one embodiment of the present invention, the primary weight value identifies a character. Furthermore, the secondary weight value specifies an accent on the character, and the tertiary weight value can specify case information for the character.

In one embodiment of the present invention, the collation element is four bytes in size, of which the primary weight field is two bytes, the secondary weight field is one byte and the tertiary weight field is one byte, unless a value in the primary weight field belongs to the reserved set of values, in which case the primary weight field takes up all four bytes of the collation element.

In one embodiment of the present invention, the reserved set of values for the primary weight value includes hexidecimal values 0xFFF0-0xFFFF.

In one embodiment of the present invention, the collation element is taken from a collation weight table that is used to map characters to collation weights in order to establish an ordering between strings of characters.

In a variation on this embodiment, the system additionally constructs a sorting key for a string by reading each character in the string and looking up a corresponding collation element for each character from the collation weight table. The system subsequently adds the corresponding collation element for each character to the sorting key. Note that if this sorting key is associated with a

record in a database, the sorting key can used to construct a linguistic index for the database.

## BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 illustrates a computer system with a database in accordance with an embodiment of the present invention.

FIG. 2 illustrates alternative structures for a collation element in accordance with an embodiment of the present invention.

FIG. 3A illustrates how a sorting key is created in accordance with an embodiment of the present invention.

FIG. 3B is a flow chart illustrating the process of creating a sorting key in accordance with an embodiment of the present invention.

FIG. 4 is a flow chart illustrating the process of reading a collation element in accordance with an embodiment of the present invention.

## DETAILED DESCRIPTION

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

The data structures and code described in this detailed description are typically stored on a computer readable storage medium, which may be any device

or medium that can store code and/or data for use by a computer system. This includes, but is not limited to, magnetic and optical storage devices such as disk drives, magnetic tape, CDs (compact discs) and DVDs (digital versatile discs or digital video discs), and computer instruction signals embodied in a transmission medium (with or without a carrier wave upon which the signals are modulated). For example, the transmission medium may include a communications network, such as the Internet.

**Computer System**

FIG. 1 illustrates a computer system 102 with a database 104 in accordance with an embodiment of the present invention. Computer system 102 can generally include any type of computer system, including, but not limited to, a computer system based on a microprocessor, a mainframe computer, a digital signal processor, a portable computing device, a personal organizer, a device controller, and a computational engine within an appliance.

Database 104 can include any type of system for storing data in non-volatile storage. This includes, but is not limited to, systems based upon magnetic, optical, and magneto-optical storage devices, as well as storage devices based on flash memory and/or battery-backed up memory. Database 104 includes a data file 106 comprised of a collection of collection of records, which are stored in insertion order. Data file 106 can be referenced through one or more indexes, such as index 108, which specifies an ordering for the records in data file 106. This ordering is typically determined by sorting an associated target column in data file 106. In order for this sorting to satisfy a specific linguistic sorting order, each character string in the target column is first converted into a sorting key by looking up characters in collation weight table 110. Note that collation weight

table 110 is simply an array that contains a collation element for each possible character.

**Structure of Collation Element**

FIG. 2 illustrates alternative structures for a collation element 204 in accordance with an embodiment of the present invention. As illustrated in FIG. 2, collation element 204 is produced by performing a lookup into collation weight table 110.

In the illustrated embodiment, collation element 204 occupies four bytes of data, and can have one of two forms. In a first form, the first two bytes of collation element 204 contain primary weight field 206, while the third byte contains secondary weight field 208 and the fourth byte contains tertiary weight field 210.

In a second form, the first two bytes of collation element 204 contain a reserved value in the range 0xFFF0-0xFFFF. This reserved value indicates that the third and fourth bytes of collation element 204 contain an extended portion of the primary weight field, instead of the secondary and tertiary weight values. In this case, the secondary and tertiary weight values are set to default values.

Note that the second form supports more than 1,000,000 different characters because each of the 16 possible values 0xFFF0-0XFFFF in the first and second bytes of collation element 204 is associated with 16 bits or 65,536 possible values in the third and fourth bytes of collation element 204.

Also note that the secondary and tertiary weight values can be set to default values because new characters with identifiers greater than 65,536 are Chinese Japanese and Korean (CJK) characters, mainly Han and Hangul Jamo characters, and there are no accent and case differences between Han/Hangul

Jamo characters. Moreover, even in Far East Asia, people always order Latin-based letters and numerical characters before CJK characters.

**Creating a Sorting Key**

FIG. 3A illustrates how a sorting key is created in accordance with an embodiment of the present invention. In FIG. 3A, a string 302 is converted character-by-character into a string of collation elements (weights) that comprise sorting key 304 by looking up the individual characters in collation weight table 110.

FIG. 3B is a flow chart illustrating the process of creating a sorting key 304 in accordance with an embodiment of the present invention. For each character 202 in a string 302, the system reads character 202 (step 306) and looks up a collation element 204 for character 202 in collation weight table 110 (step 308). The system then adds collation element 204 to sorting key 304 (step 310).

**Reading a Collation Element**

FIG. 4 is a flow chart illustrating the process of reading a collation element 204 in accordance with an embodiment of the present invention. The system starts by receiving collation element 204 during a sorting process or some other operation requiring comparisons between sorting keys (step 402). Next, the system determines if the first two (higher order) bytes of collation element 204 contain a reserved value, which is greater than or equal to 0xFFF0 (step 404). If so, the system takes the primary weight value to be all four bytes of collation element 204, and the secondary and tertiary weight values are set to default values (step 406).

If the first two bytes of collation element 204 do not contain a reserved value, the system sets the primary weight value to be the first and second bytes of

collation element 204. This is accomplished by shifting collation element 204 by 16 bits to the right, and then taking the remaining two bytes as the primary weight value. Next, the secondary weight value is taken from the third (second to lowest order) byte of collation element 204. This is accomplished by shifting collation element 204 right by eight bits and taking the secondary weight value to be the lower order byte of the remaining word. Finally, the tertiary weight value is taken from the fourth (lowest order) byte of collation element 204 (step 408).

The foregoing descriptions of embodiments of the present invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.